COMPUTER INSTITUTE MANAGEMENT SYSTEM PROJECT.

Research Proposal · March 2025

DOI: 10.13140/RG.2.2.11443.13603

1 author:

Kamal Acharya
Tribhuvan University
248 PUBLICATIONS **4,437** CITATIONS

SEE PROFILE

COMPUTER INSTITUTE MANAGEMENT SYSTEM

PROJECT BY

KAMAL ACHARYA

(Computer science and information technology, Tribhuvan University)

Date :2025/03/11

1.INTRODUCTION & OBJECTIVES OF THE PROJECT

The product INSTITUTE MANAGEMENT SYSTEM offers records of students etc. in an institute. To the users of this project are administrator, staff member. User is a person of administrative staff at an institute. Therefore "COMPUTER INSTITUTE MANAGEMENT SYSTEM" has been designed in such a way that it will automate the manual work of administrative department by maintaining records such as fee records, payroll records etc. The user can even manipulate the data such as by editing the records to update them from time to time or can delete records which are no more required.

PURPOSE OF REQUIRED DOCUMENTATION:

The software requirement specification is produced at the culmination of the analysis task. The function and performance allocated to software as a part of system engineering are refined by establishing a complete information description, a representation of system behavior, an indication of performance requirements and design constraints, appropriate validation criteria and other information pertinent to requirements.

SCOPE OF PRODUCT:

The deliverable product is named as "COMPUTER INSTITUTE MANAGEMENT SYSTEM". The scope of automation of COMPUTER INSTITUTE MANAGEMENT SYSTEM is to provide record of all students, courses, fees, payrolls etc. At an institute about their training or courses at the click of a button rather than maintaining files of papers."COMPUTER INSTITUTE MANAGEMENT SYSTEM" will automate the work of administrative staff which before this software was made was done manually. With the help of "COMPUTER INSTITUTE MANAGEMENT SYSTEM" administrative people like receptionist or center manager can get record of students at the click of a button. Record of students contains their fee status, exam result status (optional) etc.

I have designed the given proposed using visual basic 6 in the help of oracle DBA to automate the manual work of administrative department by maintaining records.

FEATURES OF COMPUTER INSTITUTE MANAGEMENT SYSTEM

- The system is secure in terms that other than authorized user, it cannot be accessed by any other person.
- User friendly interface of the software enables user to use this easily and conveniently, proper messages and guidelines have been provided as the user navigates through the forms.
- Data retrieval has been done through search forms that is user has full authority to search records based on any of the prime fields that was entered as a part of insertion module.
- Administrator can login, for safe way of purpose.
- User can delete and update records along with insertion and searching the records.
- All validations and checks have been deployed in the software to prevent entering wrong data by user by mistake. Also user can enter records by clicking the button or by pressing enter key of keyboard; similarly, esc key is used to exit from one page besides button.

Authentications

ADMIN

1. Student Entry
2. Course Entry
3. Show all student by Course
4. Fee Report
5. Instructor Entry
6. Login
7. Change Password

User / Staff Member

1. Student Entry
2. Course Entry
3. Show all student by Course
4. Fee Report
5. Instructor Entry
6. Login
7. Change Password

Establish the need of new system

1. Problem of Reliability: Current system is not reliable. It seems to vary in quality from one month to the, next. Some times it gives good output, but some times the output is worst.

2. **Problem of Accuracy**: There are too many mistakes in reports.

3. Problem of timeliness: In the current system the reports and output produced is mostly late and in most of the cases it is useless because it is not on time.

4. Problem of Validity: The output and reports mostly contains misleading information. The customer's information is sometimes not valid.

5. Problem of Economy: The current system is very costly. We have to spend lots of money to keep the system up and going, but still not get the desired results.

6. Problem of Capacity: The current system is suffering from problem of capacity also. The staff for organization is very less and the workload is too much. Few peoples cannot handle all the work.

1.2 Proposed System

l. Students Details: The new proposed system stores and maintains all the student details.

2. Cources Details: The new proposed system stores and maintains all types of cources details.

3. Registers: There is no need of keeping and maintaining records register manually. It remembers each and every record and we can get any report related to students at any time.

4. Speed: The new proposed system is very fast with 100% accuracy and saves time.

5. Manpower: The new proposed system needs less manpower. Less people can do the large work.

6. Efficiency: The new proposed systems complete the work of many sales person in less time.

8. Reduces redundancy: The most important benefit of this system is that it reduces the redundancy of data within the data.

9. Security : Security has been provided to prevent unauthorized access.

1.3 SOFTWARE & HARDWARE REQUIREMENTS

The need of today's software development is competence in a GUI based front-end tool, which can connect to Relational Database engines. This gives the programmer the opportunity to develop client server based commercial applications.

FRONT END

Visual Studio programming tools are complete programming environments. It allows programmers to build a GUI program using the various on-screen controls such as buttons, text, menus, boxes etc. These controls are placed on a form and then the processing details related with each control are filled in.

In the business world, competitive strategies have become the order of the day to improve quality, cut costs and provide a high response customer service base. Most organizations today need to be market driven and do a lot of value addition to their products and services. This naturally calls for rational decision making, which requires information. Information Technology or IT provides that effective channel to support and implement this strategy. Technology that empowers the desktop, thus setting a trend for the way successful organizations will use technology in the next decade.

BACK END

In oracle the logical and physical structures are separate and thus this helps in data independence i.e., physical storage of data can be managed without affecting the access to logical storage structures. Oracle is a major database which along with its added features passes the ACID test, which is important in insuring the integrity of data. Oracle is very much suited for enterprise computing and large database applications.

There are various branches in which one can go into by learning oracle like be a functional consultant, or a Database administrator DBA in short, Oracle PL/SQL Developer Certified Associate designation, Oracle Forms Developer Certified Professional designation and so on.

2.1 Preliminary Investigation :

System development, a proccess consisting of two major steps of system analysis and design, start when management or sometimes system development personnel feel that a new system or an improvement in the existing system is required. The sustem development life cycle is classically thought of as the set of activities that analysts, desiners and users carry out to develop and implement an information system. The system development life cycle consists of the following activities:

- Preliminary investigation
- Determination of system requirements
- Design of system
- Development of software
- System testing
- Implementation, evaluation, and maintenance

A request to take assistance from information system can be made for many reasons, but in each case someone in the organisation initiates the request is made, the first system activity the preliminary investigation begins. This activity has three parts:

1) Request clarification
2) Feasibility study
3) Request approval

Request clarification: Many requests from employees and users in the organisations are not clearly defined, Therefore it becomes necessary that project request must be examined and clarified properly before considering systems investigation.

2.2 SYSTEM DEVELOPMENT LIFE CYCLE

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, The subject System Analysis and Design (SAD), mainly deals with the software development activities.

DEFINING A SYSTEM

A collection of components that work together to realize some objective forms a system. Basically there are three major components in every system, namely input, processing and output.

In a system the different components are connected with each other and they are interdependent. For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system and so forth. The objective of the system demand that some output is produced as a result of processing the suitable inputs.

SYSTEM LIFE CYCLE

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system.

System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the System Analysis and Design terminology, the system development life cycle means software development life cycle.

Following are the different phases of software development cycle:

- System study
- Feasibility study
- System analysis
- System design
- Coding
- Testing
- Implementation
- Maintenance

The different phases of software development life cycle is shown below.

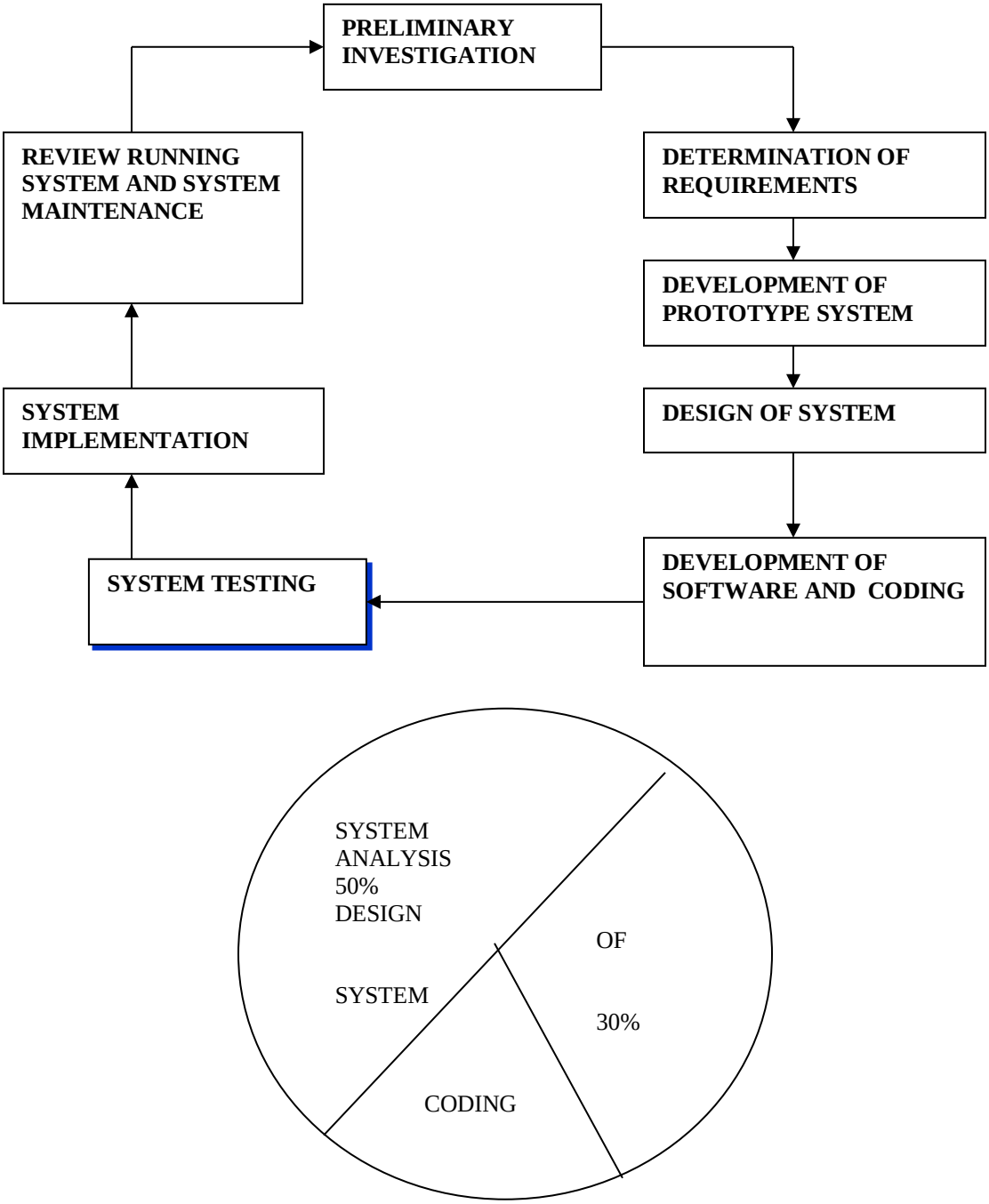


FIG: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME DEVOTED

2.3 FEASIBILITY STUDY

The feasibility study proposes one or more conceptual solution to the problem set of the project. In fact, it is an evaluation of whether it is worthwhile to proceed with project or not.

Feasibility analysis usually considers a number of project alternatives, one that is chosen as the most satisfactory solution. These alternatives also need to be evaluated in a broad way without committing too many resources. Various steps involved in feasibility analysis are:

1. To propose a set of solution that can realize the project goal. These solutions are usually descriptions of what the new system should look like.

2. Evaluation of feasibility of such solutions. Such evaluation often indicates shortcomings in the initial goals. This step is repeated as the goals are adjusted and the alternative solutions are evaluated.

COST AND BENEFIT ANALYSIS

Developing an IT application is an investment. Since after developing that application it provides the organization with profits. Profits can be monetary or in the form of an improved working environment. However, it carries risks, because in some cases an estimate can be wrong. And the project might not actually turn out to be beneficial.

Cost benefit analysis helps to give management a picture of the cost, benefits and risks. It usually involves comparing alternate investments.

Cost benefit determines the benefits and savings that are expected from the system and compares them with the expected costs.

In performing cost and benefit analysis it is important to identify cost and benefits factors. Cost and benefits can be categorized into the following categories:

1. **Development Costs –** Development costs is the costs that are incurred during the development of the system. It is one time investment.
2. **Operating Costs –** Operating Costs are the expenses required for the day to day running of the system. Examples of Operating Costs are Wages, Supplies and Overheads.
3. **Hardware/Software Costs –** It includes the cost of purchasing or leasing of computers and it's peripherals. Software costs involves required S/W costs.
4. **Personnel Costs –** It is the money spent on the people involved in the development of the system**.**
5. **Facility Costs –** Expenses that are incurred during the preparation of the physical site where the system will be operational. These can be wiring, flooring, acoustics, lightning, and air-conditioning.
6. **Supply Costs –** These are variable costs that are very proportionately with the amount of use of paper, ribbons, disks, and the like.

- **BENEFITS**

We can define benefits as

Profit or Benefit = Income – Costs

Benefits can be accrued by:

- Increasing income, or
- Decreasing costs, or
- Both

SYSTEM OVERVIEW

The limited time and resources have restricted us to incorporate, in this project, only a main activities that are performed in an Online Institute Management System, but utmost care has been taken to make the system efficient and user friendly. “Institute Management System” has been designed to computerized the following functions that are performed by the system:

1. New records are entered.
2. Present record can be updated.
3. Record not needed can be deleted.
4. Existing record can be searched.
5. Prevents unauthorized access.

3.1 IMPORTANCE OF COMPUTERIZED

There are several attributes in which the computer based information works. Broadly the working of computer system is divided into two main groups:

- Transaction System
- Decision Support System

Transaction System:

A transaction is a record of some well-defined single and usually small occurrence in a system. Transactions are input into the computer to update the database files. It checks the entering data for its accuracy. This means that numeric data appears in numeric field and character data in character field. Once all the checks are made, transaction is used to update the database. Transaction can be inputted in on-line mode or batch mode. In on-line mode, transactions are entered and updated into the database almost instantaneously. In batch mode, transactions are collected into batches, which may be held for a while and inputted later.

Decision Support System:

It assists the user to make analytical decision. It shows the various data in organized way called analysis. This analysis can be made to syrdy preferences and help in making decisions.

Computer system works out best with record maintenance. It will tell you which customer would get how much pending/reports statements. It will also help to search the information about a particular person by simply entering his telephone number. User can store information as per requirement, which can be used for comparison with other reports.

SPECIFIC REQURIMENTS:

New records are entered through the graphical user interface i.e the forms and at the click of a button they are entered into the database, so that they can be retrieved for future use or any enquiry.

EXTERNAL INTERFACE REQUIREMENTS:

User Interface : "Computer Institute Management System**"** provides complete graphical user interface which will make software more interactive & convenient to use.

Hardware Interface: "Computer Institute Management System" will interact with almost all types of hardware available at an institute.

Software Interface: Software uses VB.Net for forms designing and to provide good graphical user interface. Oracle DBA & SQL Server is used at the backend to store data.

NON FUNCTIONAL REQUIREMENT:

Performance Requirement: Using VB.Net with the helpe of oracle DBA & SQL Server enhances the performances of project as both are Microsoft products therefore front and backend have been chosen such that the performance increases.

Software Quality Attributes: "Computer Institute Management System" with stands following quality attributes: correctness in terms of number of computational errors, Robustness.

3.2 FUNCTIONAL REQIREMENTS:

Functional Module 1: To login and password facility has been provided to authenticate the user.

Input: password provided by administrator.

Process: designing the login form.

Functional Module 2: User enters the data through forms the database.

Purpose: Data is required for later use by the institute also to keep records of students performance.

Input: student that come for courses or training in the institute register themselves.

Process: designing the forms and entering data into database.

Functional Module3: To enter and maintain projects being made t an institute.

Input: Input for these forms is provided to user by the group leader and guide of the project.

Process: Designing the project record entry form.

Functional Module4: To enable user to edit data

Purpose: Data or information changes with time therefore records need to be updated.

Input: monthly assessment of each student

Process: Updating database with new records

Functional Module5: To delete records that is no more required

Purpose: Saving space and memory requirements and thus increasing accessibility and retrieval of data faster.

Input: Studetn ID no of student or primary key of the table.

Process: updating database.

Functional Module6: Retrieval of data

Purpose: user needs for future planning or for studying a student's performance graph.

Input: primary key of tables from which we want to extract data.

Process: generating reports.

Functional Module7: Fee Details

Purpose: Student, Parents needs to get their updated student's Fee Details. Administrator add these details.

Input: primary key of tables from which we want to extract data.

Process: generating reports.

3.3. DATA DESCRIPTION

Data entity 1:

First thing is the login and the password facility, it has been provided to authenticate the user that is other than administrative staff like students etc should not have access to the records as they may try to change the records like fee and attendance status.

Data entity 2:

Second important information is registration form, fee record form, attendance form and lastly the accessories form. These forms appear again for different category of students at an institute that allows i.e. one, who come for industrial training/ project in their final semesters, secondly those who join an institute with purpose of doing the courses offered by it, in this case one student can opt for more than one course thus he has been categorized as new and existing student, finally those customers who are or were employees of some organization and have joined an institute under some scheme like one for VRS employees.

Data Entity 3:

Third important entity is the data retrieval forms that take input as user parameters for the search and generate report for the same.

DEPENDENCY DESCRIPTION

Module Dependency:

All the modules are interrelated as editing; deletion and retrieval can be performed only after successful execution of the data entry module, same way unless and until login and password form is executed user cannot enter the options page.

Process Dependency:

Interdependence of validation process, entering records process, edit, delete and retrieval process & all its sub processes. All the processes are dependent upon the validation process as in if a user does not surpasses this process the user will not be able to move onto other processes.

Data Dependency:

String of characters entered as password, allows user to move to other page, & options are displayed. Thus, validation module, basically deals with data i.e. password.

Registration form entries act as input to databases that keep record of new & existing students. Information like name of student, data of birth etc entered by user is entered in database all this data is the part of edit, delete & retrieve module of Institute Management System.

Test Data:

The user at the time of login, if enters wrong password then appropriate message will be flashed & user stays on the same form.

E.g. Say password is “Institute Management System” & if user enters some other string then error will be displayed as invalid password.

If wrong data is entered in text boxes during the filling of Registration form then as the user clicks “OK” button valid types will appear & invalid fields will be made to fill again.

System Design

The design document that we will develop during this phase is the blueprint of the software. It describes how the solution to the customer problem is to be built. Since solution to complex problems isn't usually found in the first try, iterations are most likely required. This is true for software design as well. For this reason, any design strategy, design method, or design language must be flexible and must easily accommodate changes due to iterations in the design . Any technique or design needs to support and guide the partitioning process in such a way that the resulting sub-problems are as independent as possible from each other and can be combined easily for the solution to the overall problem. Sub-problem independence and easy combination of their solutions reduces the complexity of the problem. This is the objective of the partitioning process. Partitioning or decomposition during design involves three types of decisions: -

Define the boundaries along which to break;

Determine into how money pieces to break; and

Identify the proper level of detail when design should stop and implementation should start.

Basic design principles that enable the software engineer to navigate the design process suggest a set of principles for software design, which have been adapted and extended in the following list:

Free from the suffer from "tunnel vision." A good designer should consider alternative approaches, judging each based on the requirements of the problem, the resources available to do the job.

The design should be traceable to the analysis model. Because a single element of the design model often traces to multiple requirements, it is necessary to have a means for tracking how requirements have been satisfied by the design model.

The design should not repeat the same thing. Systems are constructed using a set of design patterns, many of which have likely been encountered before. These patterns should always be chosen as an alternative to reinvention. Time is short and resources are limited! Design time should be invested in representing truly new ideas and integrating those patterns that already exist.

The design should "minimize the intellectual distance" between the software and the problem as it exists in the real world. That is, the structure of the software design should (whenever possible) mimic the structure of the problem domain.

The design should exhibit uniformity and integration. A design is uniform if it appears that one person developed the entire thing. Rules of style and format should be defined for a design team before design work begins. A design is integrated if care is taken in defining interfaces between design components.

The design of a system is essentially a blueprint or a plan for a solution for the system. We consider a system to be a set of components with clearly defined behavior that interacts with each other in a fixed defined manner to produce some behavior or services for its environment. A component of a system can be considered a system, with its own components. In a software system, a component is a software module.

The design process for software systems, often, has two levels. At the first level, the focus is on deciding which modules are needed for the system, the specifications of these modules, and how the modules should be interconnected. This is what is called the system design or top-level design. In the second level, the internal design of the modules, or how the specifications of the module can be satisfied, is decided. This design level is often called detailed design or logic design. Detailed design essentially expands the system design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding.

Because the detailed design is an extension of system design, the system design controls the major structural characteristics of the system. The system design has a major impact on the testability and modifiability of a system, and it impacts its efficiency. Much of the design effort for designing software is spent creating the system design.

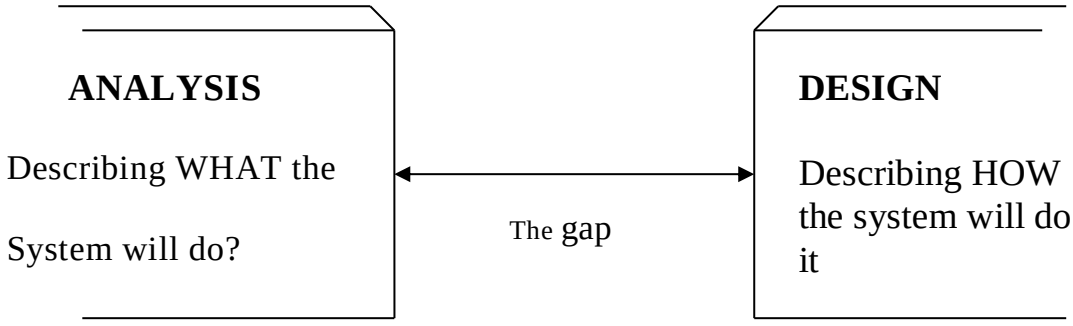
The input to the design phase is the specifications for the system to be designed. Hence, reasonable entry criteria can be that the specifications are stable and have been approved, hoping that the approval mechanism will ensure that the specifications are complete, consistent, unambiguous, etc.

The final deliverable from system analysis is a document containing an unambiguous statement of the client's requirements from the new system and what the development project will have to deliver in order to be considered a success.

The functional specification is the starting point for designing, which depends to a large extent on the accuracy and thoroughness with which the analysis has been carried out. Understanding of the business, appreciation of the client's problems and documentation of requirements provide the foundation on which the designing is based.

A key factor in this approach is the use of the structured techniques.

Analysis ends with a description of what the new system must do to fulfill the requirements of the organization, while design specifies how this will be done by selecting one of the many ways of doing it.



The structured techniques used during the analysis which provide this logical view are:

• Data flow diagrams - representing the process, which manipulate the data as it passes through the system.

• Entity model - showing the relationship within the data items held within the system.

• A data dictionary - providing an overall consistent definition of the data used during the system development. This definition includes the content of the data stores, data flows and the process shown on the data flow diagrams, and the entities that make up the entity model.

DESIGN OBJECTIVES AND CONSTRAINTS:

Flexible –

The design would enable future requirements of the organization to be incorporated without much difficulty. Often the organizational needs and objectives change over time and hence such a design enables the system to reflect these changes.

• Maintainable -

A good design is easy to maintain and this reduces the client's maintenance cost, which usually represents a proportion of the lifetime of the system.

• Portable -

A client for whom the software was developed may wish to change the hardware on which the system run. A good design is portable - in other words it is capable of being transferred from one machine environment to another with minimum amount of effort.

• Easy to use -

With increasing number of general users having exposure to computers and access to web sites, expectations of computer applications in term of their ease of use are also increasing. A good design will result in a system which is `user - friendly' - easy to understand, not difficult to learn how to use and straightforward to operate.

• Reliable -

The system designed must be secure against human error, deliberate misuse or machine failure, and which the data will be stored without corruption.

4.1 PHYSICAL DESIGN

The design phase focuses on the detailed implementation of the system recommended in the feasibility. Emphasis is on translating performance specifications into design specifications. The design phase is a transition from user-oriented document to a programmer-oriented document.

a. Design Methodology:

Design Methodology is a way to transform the "art" of system analysis and design into an "engineering - type" discipline. It explains the relationship amongst various modules and programs with in the system. It standardizes the approach to analysis and design, simplifies design by segmentation, improves documentation and subsequent maintenance and enhancements.

The following structured diagram can appropriately represent the relationship between various modules .

b. Design Overview:

In analyzing the present system a great deal of information was collected during the investigation and feasibility phases through list of problems and requirements, interview reports, questionnaires, onsite observations, manuals and determining potential solutions.

It is important to record this information in an unambiguous, concise manner which will be clear and accessible to others, and which can be used by other analysts and designers involved in developing the system. Structured techniques help us to record the information in this way, using diagrams and minimum amount of the text.

Structured analysis is a set of techniques and graphical tools that allow the analyst to develop a new kind of system specification that are easily understandable to the user. The traditional approach of organizing data through flowcharts support future

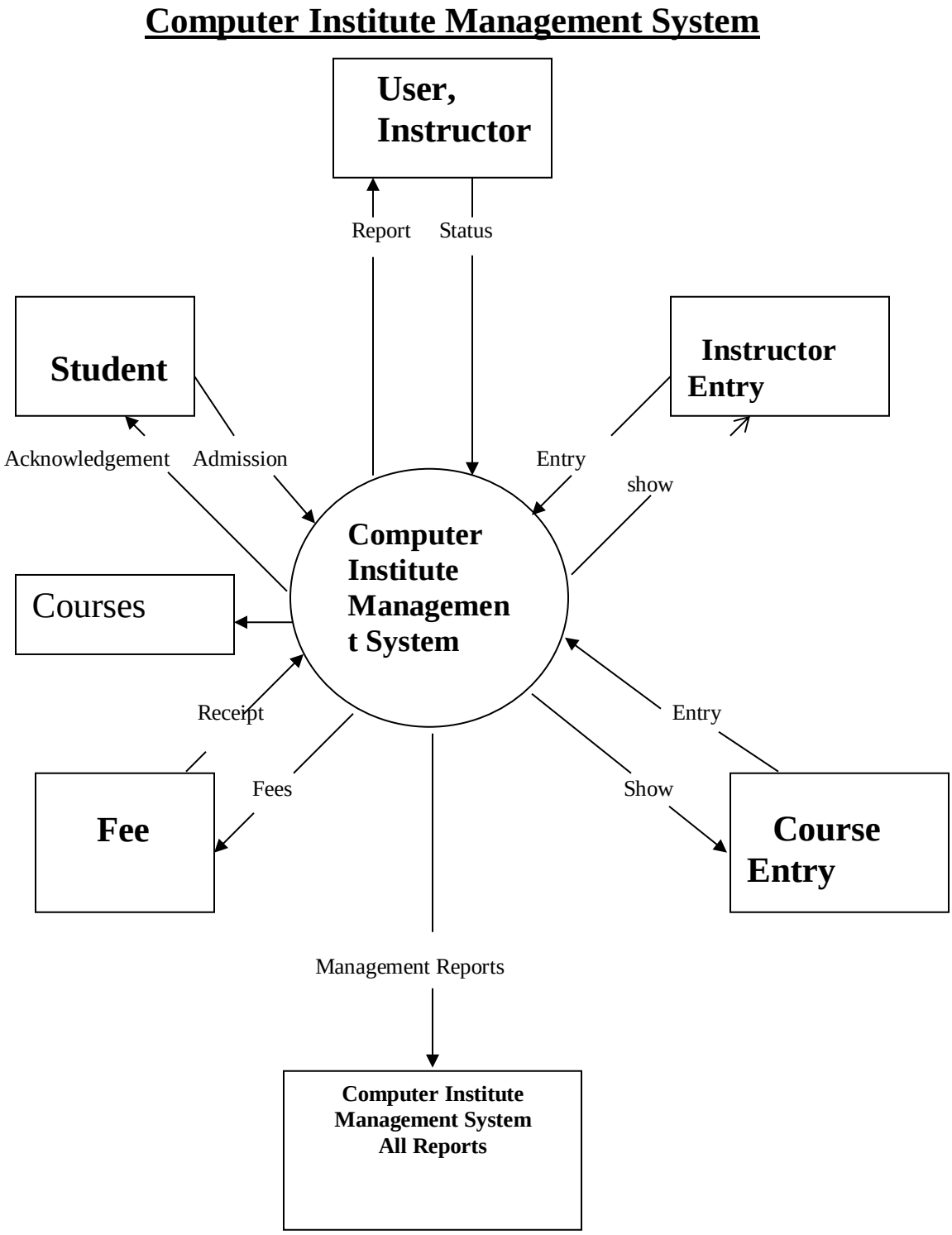
developments and simplify communication with the user but focus on the cost/benefit and feasibility analysis, project management, hardware and software selection, and personal considerations. In contrast, structured analysis considers new goals and structured tools for

analysis, which provide the basis for design and implementation.

c. Process Modeling:

System design goes through two phases of development: logical and physical. Logical implementation represented by Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system it describes the input (source), outputs (destinations), data bases (data stores), and procedures (data flows) - all in the format that meets the user's requirements. The logical implementation of the whole project can be represented as under through Data Flow Diagrams (DFD).

4.2 ER diagrams



4.3 **Process Logic for Each Module:**

Validation: This login and password facility has been provided to authenticate the user. The process is designing the login form.

Record Entry: Entering the records via forms designed for user convenience and finally records are stored into the database.

Editing the records: The process is to enable user to edit. Data or information changes with time therefore records need to be updated like monthly assessment of each student etc.

Deleting the Records: The process is to delete records that are no more required. Purpose is to save space and memory requirement and thus increasing accessibility and retrieval of data faster.

4.4 Data Modeling

Data Structure Requirements according to the modules:

- Identify The Various Tables Required.
- Fields for These Tables.
- The Various Key Fields (for example Primary key and foreign key).
- Identify The Various Constraints like Not Null, Unique etc.

Describing Data Elements:

Each entry in the data dictionary consists of a set of details describing the data used or produced in the system. Each item is identified by a data name, description, alias, and length and has specific values that are permissible for it in the system being studied.

4.5 SCHEDULING

Scheduling of a software project does not differ greatly from scheduling of any multi- task engineering effort. Therefore, generalized project scheduling tools and techniques can be applied with little modification to software projects.

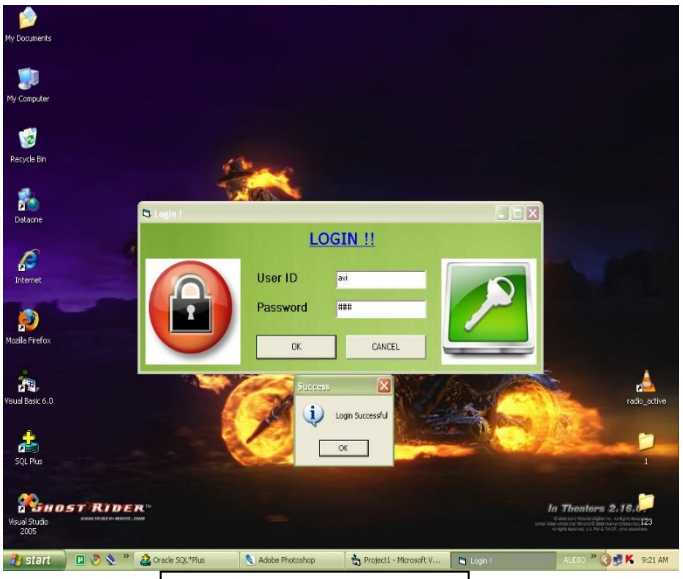
Program evaluation and review technique (PERT) and critical path method (CPM) are two project scheduling methods that can be applied to software development. Both techniques are driven by information already developed in earlier project planning activities.

Estimates of Effort

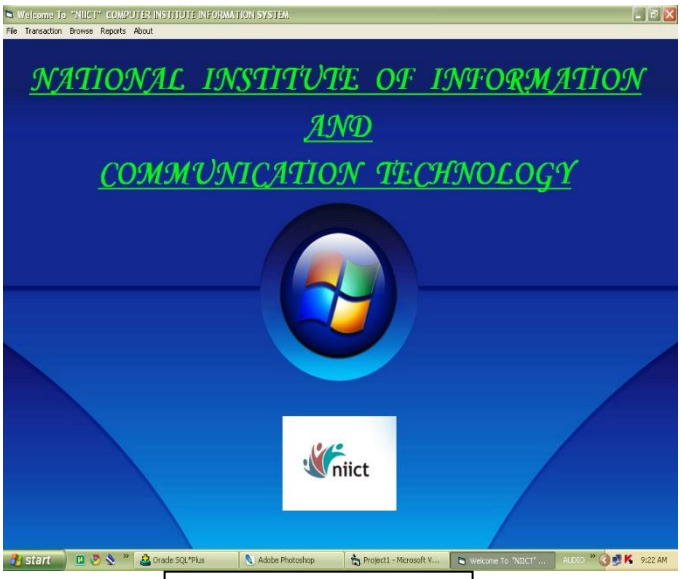
- A decomposition of the product function.
- The selection of the appropriate process model and task set.
- Decomposition of tasks.

Both PERT and CPM provide quantitative tools that allow the software planner to (1) determine the critical path-the chain of tasks that determines the duration of the project; (2) establish "most likely" time estimates for individual tasks by applying statistical models; and (3) calculate "boundary times" that define a time window" for a particular task.

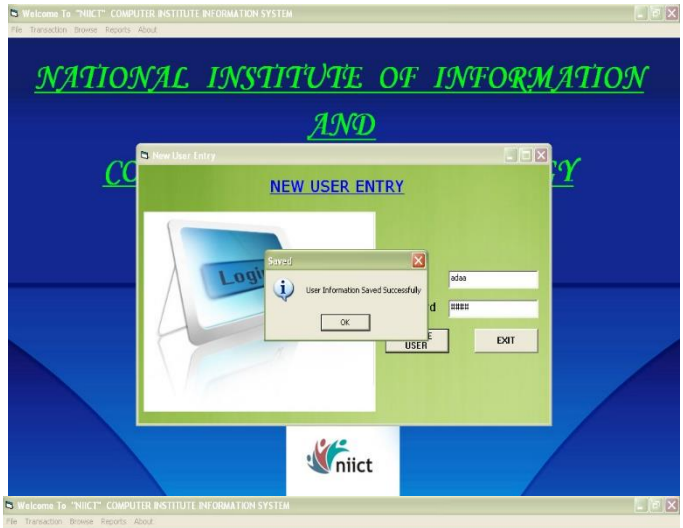
Boundary time calculations can be very useful in software project scheduling. Slippage in the design of one function, for example, can retard further development of other functions. It describes important boundary times that may be discerned from a PERT or CPM network: (I) the earliest time that a task can begin when preceding tasks are completed in the shortest possible time, (2) the latest time for task initiation before the minimum project completion time is delayed, (3) the earliest finish-the sum of the earliest start and the task duration, (4) the latest finish- the latest start time added to task duration, and (5) the total float-the amount of surplus time or leeway allowed in scheduling tasks so that the network critical path maintained on schedule. Boundary time calculations lead to a determination of critical path and provide the manager with a quantitative method for evaluating progress as tasks are completed.



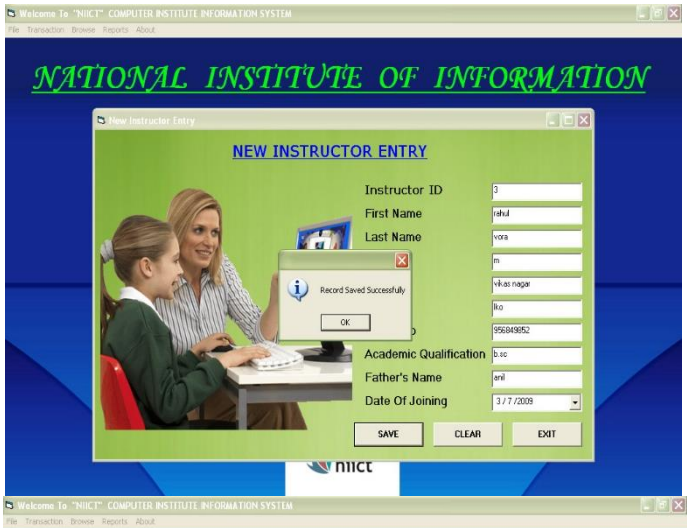
Login Window



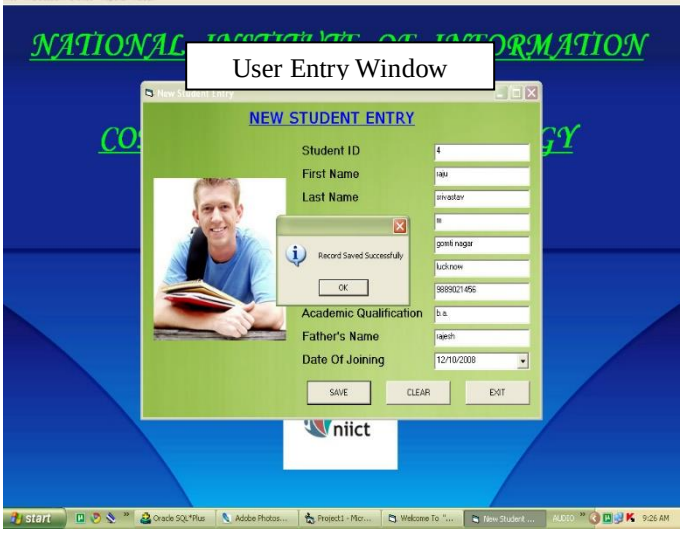
Main Window



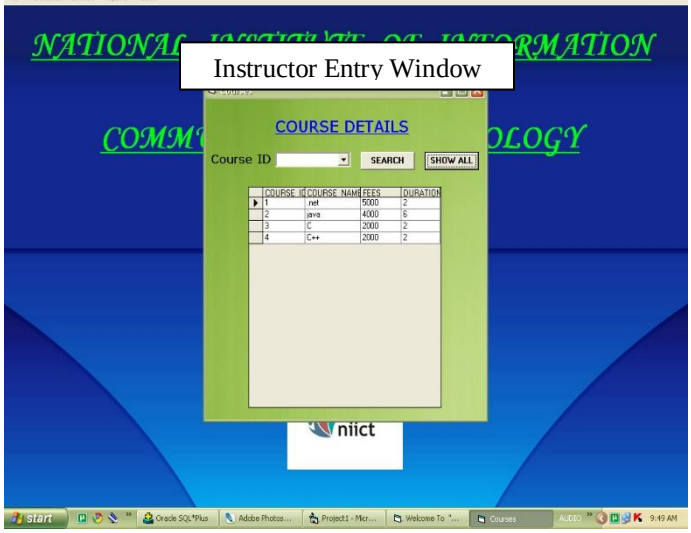
User Entry Window



Instructor Entry Window



Student Entry Entry



Course View Window

5.2 Code Efficiency

Reviewing of Code efficiency for a module is carried out after the module is successfully compiled and all the syntax errors eliminated. Code efficiency review is extremely cost-effective strategies for reduction in coding errors in order to produce high quality code. Normally, two types of efficiency are carried out on the code of a module - code optimization and code inspection. The procedure and final objective of these two efficiency techniques are very different as discussed below.

5.3 Optimization of Code

Code optimization is an informal code analysis technique. In this technique, after a module has been coded, it is successfully compiled and all syntax errors are eliminated. Some members of the development team are given the code a few days before the optimization meeting to read and understand the code. Each member selects some test cases and simulates execution of the code by hand (i.e. trace execution through each statement and function execution). The main objectives of the optimization are to discover the algorithmic and logical errors in the code. The members note down their findings to discuss these in a optimization meeting where the coder of the module is also present.

Even though a code optimization is an informal analysis technique, several guidelines have evolved over the years for making this naïve technique more effective and useful. Of course, these guidelines are based on personal experience, common sense, and several subjective factors. Therefore are based on personal experience, common sense, and several subjective factors. Therefore, guidelines should be considered as examples rather than as rules to be applied dogmatically. Some of these guidelines are the following:

The team performing the code optimization should not be either too big or too small. Ideally, it should consist of three to seven members.

6.1 Testing Phase

One of the purposes of the testing is to validate and verify the system. Verification means checking the system to ensure that it is doing what the function is supposed to do and Validation means checking to ensure that system is doing what the user wants it to do.
No program or system design is perfect; communication between the user and the designer is not always complete or clear, and time is usually short. The result is errors and more errors. Theoretically, a newly designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the user's requirements. This is the best chance to detect and correct errors before the system is implemented. The purpose of system testing is to consider all the likely variations to which it will be subjected and then push the system to its limits. If we implement the system without proper testing then it might cause the problems.

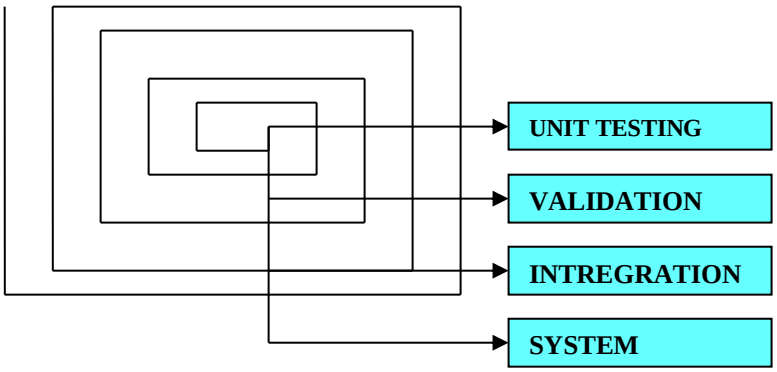
1. Communication between the user and the designer.
2. The programmer's ability to generate a code that reflects
   exactly the system specification.
3. The time frame for the design.

Theoretically, a new designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the requirements of the user.

The process of system testing and the steps taken to validate and prepare a system for final implementation are:

LEVELS OF TESTING:

The different types of testing are as follows:



1**. Unit Testing**:

This is the smallest testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests.

2. **Integration Testing**:

In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top-down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black box testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

3. **System Testing**:

A part from testing the system to validate the functionality of software against the requirements, it is also necessary to test the non-functional aspect of the system. Some examples of non-functional tools include tests to check performance, data security, usability/user friendliness, volume, load/stress that we have used in our project to test the various modules.

System testing consists of the following steps:

1. Program(s) testing.
2. String testing.
3. System testing.
4. System documentation.
5. User acceptance testing.

4. **Field Testing**:

This is a special type of testing that may be very important in some projects. Here the system is tested in the actual operational surroundings. The interfaces with other systems and

the real world are checked. This type of testing is very rarely used. So far our project is concerned, we haven't tested our project using the field testing.

5. **Acceptance Testing**:

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and re-tests the system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user. In this project, it is the job of the bookstores to check the system that whether the made system fulfills the goals or not.

Why System Testing?

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct , the goal will be successfully achieved. Inadequate testing results in two type of problems:

1. The time lag between the cause and the appearance of the problem.
2. The effect of system errors on the files and records within the system.

Prepare Test

A workable test plan must be prepared in accordance with established design specifications. It includes the following items:

• Outputs expected from the system.
• Criteria for evaluating outputs.
• A volume of test data.
• Procedure for using test data.
• Personnel and training requirements.

Specify Conditions for User Acceptance Testing

Planning for user acceptance testing calls for the analyst and the user to agree on conditions for the test.

Prepare Test Data for Program Testing

As each program is coded, test data are prepared and documented to ensure that all aspects of the program are properly tested.

Prepare Test Data for Transaction Path Testing

This activity develops the data required for testing every condition and transactions to be introduced into the system. The path of each transaction from origin to destination is carefully tested reliable results.

Compile / Assemble Programs

All programs have to be compiled / assembled for testing.

Systems testing

The computer department to ensure that the system functions as specified does this testing. This testing is important to ensure that a working system is handed over to the user for acceptance testing.

Acceptance testing.

The user to ensure that the system functions, as the user actually wanted performs this testing. With prototyping techniques, this stage becomes very much a formality to check the accuracy and completeness of processing. The screen layouts and output should already have been tested during the prototyping phase.

An error in the program code can remain undetected indefinitely. To prevent this from happening the code was tested at various levels. To successfully test a system, each condition, and combinations of conditions had to be tested. Each program was tested and linked to other programs. This unit of program is tested and linked to other units and so on until the complete system has been tested.

The purpose of testing is to ensure that each program is fully tested. To do so a test plan had to be created. The test plan consists of a number of test runs such as the valid paths through the code, and the exception and error handling paths. For each test run there is a list of conditions tested, the test data used and the result expected. The test plan was then reviewed to check that each path through the code is tested correctly. It is the responsibility of the programmer to collect the data that will produce the required test condition.

6.2 **Verification and Validation (V&V)**

The objectives of verification, validity activities are to assess and improve the quality of the work products generated during development and modification of the software. Quality depends upon the various attributes like correctness, completeness, consistency, reliability, usefulness, usability, efficiency and conformance to standards.

Verification activities include proving, testing, and reviews. Validation is the process of evaluating software at the end of the software development to ensure compliance with the software requirements. Testing is a common method of validation. Clearly, for high reliability we need to perform both activities. Together, they are often called V&V activities.

The major V&V activities for software development are inspection, reviews, and testing (both static and dynamic). The V&V plan identifies the different V&V tasks for the different phases and specifies how these tasks contribute to the project V&V goals. The methods to be used for performing these V&V activities, the responsibilities and milestones for each of these activities, inputs and outputs for each V&V task, and criteria for evaluating the outputs are also specified.

The two major V&V approaches are testing and inspections. Testing is an activity that can be generally performed only on code. It is an important activity and is discussed in detail in a later chapter. Inspection is a more general activity that can be applied to any work product, including code. Many of the V&V tasks are such that for them, an inspection type of activity is the only possible way to perform the tasks (e.g. trace ability and document evaluation). Due to this, inspections play a significant role in verification.

6.3 System Implementation Maintenance and Review

As we know, creating software is one thing and the implementation of the created software is another. The process of implementing software is much difficult as compared to the task of creating the project. First we have to implement the software on a small scale for removing the bugs and other errors in the project and after removing them we can implement the software on a large scale.

Before we think in terms of implementing the Software on a large basis, we must consider the Hardware requirements.

Whenever we develop a software or project a certain hardware and software is being used by the programmer for developing the project. The hardware and software to be used by the programmer for developing the project should be such that it would result in the development of a project, which would satisfy all the basic needs for which the project has been created by the programmer. The Hardware should be such that cost constraints of the Client should also be taken into account without affecting the performance.

HARDWARE EVALUATION FACTORS

When we evaluate computer hardware, we should first investigate specific *physical and performance* characteristics for each hardware component to be acquired. These specific questions must be answered concerning many important factors. These *hardware evaluation factors* questions are summarized in the below figure.

Notice that there is much more to evaluating hardware than determining the fastest and cheapest computing device. For e.g. the question of possible obsolescence must be addressed by making a technology evaluation. The factor of *ergonomics* is also very important. Ergonomics is the science and technology that tries to ensure that computers and other technologies are "user-friendly", that is safe, comfortable and easy to use. *Connectivity is* another important evaluation factor, since so many computer systems are now interconnected within wide area or local area telecommunications networks.

HARDWARE EVALUATION FACTORS

1) Performance
2) Cost
3) Reliability
4) Availability
5) Compatibility
6) Modularity
7) Technology
8) Ergonomics
9) Connectivity
10) Environmental requirements
11) Software
12) Support

SOFTWARE EVALUATION FACTORS

Software can be evaluated according to many factors similar to the hardware evaluation. Thus the factors of *performance, cost, reliability, compatibility, modularity, technology, ergonomics, and support* should be used to evaluate proposed software acquisitions. In addition, however, *the software evaluation factors* are summarized in below figure. For e.g. some software packages require too much memory capacity and are notoriously slow, hard to use, or poorly documented. They are not a good selection for most end users, even if offered at attractive prices.

SOFTWARE EVALUATION FACTORS:

1) Efficiency: is the software a well-written system of computer instructions that does not use much memory capacity or CPU time?
2) Flexibility: can it handle its processing assignments easily without major modifications?
3) Security: does it provide control procedures for errors, malfunctions and improper use?

4) Language: do our computer programmers and users write it in a programming language that is used?

5) Documentation: is the s/w well documented? Does it include helpful user instructions?

6) Hardware: does existing hardware have the features required to best use this software?

7) Other characteristics of hardware such as its performance, what about the cost, how much is reliable and etc.

6.4 SECURITY MEASURES

User Name & Password security implemented so that nounauthorised person can handle any operation without username and Password.

- Only authorized person can log-on the system.
- Only authorized person can update the records.
- Only authorized person can handle the reservation.
- Only authorized person can print the report.

It has two kinds of users:

1. Administrator
2. User1

Administrator: He has complete authority (Read, Add, Modify) of operating the software. The User Name and Password provided to the Administrator in this project is:

User Id: Admin
Password: admin123

User1: When this user logs onto the system, he can only view information and other reports. He can generate different reports.

User Id: User1
Password: user123

7. COST ESTIMATION OF THE PROJECT

Cost in a project is due to the requirements for software, hardware, and human resources. Hardware resources are computer time, terminal time and memory required for the project. Software resources include the tools and compilers needed during development. The bulk of cost of software development is due to human resources needed. Cost estimates are determined in terms of person-months **(PM)**.

Total No. Of Persons Involved In This Project:

1 Administrator

2 Senior Programmer

3 Junior Programmers

Since this Project will complete in 3 months

Cost Estimate: (Salary of Project Manager + Salary of Senior Programmer + 2 * Salary of Junior Programmer) * 2

Further Scope of the Application

1. Though maximum efforts have been put in to make this report authentic in all aspects and to take all necessary presentation to ensure that the information gathered is true, some uncomfortable factors may have crept in.

2. Some of the respondents were reluctant to part with certain information on the pretext of the sensitivity of the information. Also some facts of figures were not divulged as the company policy came in the way for free revelation of the desired input.

3. An element of bias might have crept in from the side of the official interviewed. This could also have resulted in some kind of modification of the information divulged.

4. Through an attemptwas make to collect information from the best possible source in the company, it was difficult to meet the top officials due to their busy schedules.

5. Most of the analysis and interpretations, made for this report, are based on secondary data obtained. This data could have some inherent mistakes and errors.

6. Finally, although due care has been taken those can be typing and compilation errors in the report itself.

- The tasks specified were not well defined because nothing was mentioned regarding validations in the project. Though gave maximum effort to check the software with different validation tests, a few of them might be present in this version

- Though all possible checks have been placed but still there are some controls where checks have not been deployed, therefore user has to be careful while entering data through these controls. For e.g.: data entry has to be in (mm/dd/yyyy) format.
- Screen resolution has to be 1024 x 768 otherwise forms will not be visible
- Due to limited time available survey could not be undertaken for intended 20 consumers and thus had to be limited to 10
- Communication gaps exist between employees and management, as seniors don't share problem with subordinates resulting in violation of psychological contract.
- Poor rewarding system(slow)
- Poor working conditions

The limitations may be many and the magnitude of the influence of these limiting factors may have a bearing on the report, but it in no way alters the ultimate aim of the project and because it's highly USER FRIENDLY, it would be the choice of all kinds of personnel.

8. CONCLUSIONS

This project has been a rewarding experience in more than one way. The entire project work has enlightened us in the following areas.

- We have gained an insight into the working of the Hospital. This represents a typical real world situation
- Our understanding of database design has been strengthened this is because in order to generate the final reports of database designing has to be properly followed.
- Scheduling a project and adhering to that schedule creates a strong sense of time management.
- Sense of teamwork has developed and confidence of handling real life project has increased to a great extent.
- Initially, there were problem with the validation but with discussions , we were to implement validations.

9. Bibliography

- VB.NET — Jamew Foxall
- VB.NET Projects — Tony Martin,Dominic Selly
- VB.Net Professional Projects — Kuljit Kaur,Pooja Bembey
- ORACLE 9 The Complete Reference — George Koch Kevin Loney

10.GLOSSARY

Text Box

A text Box control, sometimes called an edit field or edit control, displays information entered at design time, entered by the user, or assigned to the control in code at run time.

Button

Use a Command Button control to begin, interrupt, or end a process. When chosen, a Command Button appears pushed in and so is sometimes called a push button.

List Box

A List Box control displays a list of items from which the user can select one or more. If the number of items exceeds the number that can be displayed, a scroll bar is automatically added to the List Box control.

Label

A Label control is a graphical control you can use to display text that a user can't change directly.

GROUPBOX

A GROUPBOX control provides an identifiable grouping for controls. You can also use a Frame to subdivide a form functionally – for example, to separate groups of Option Button controls.

Combo Box

A Combo Box control combines the features of a text box and a list box. This control allows the user to select an item either by typing text into the combo box, or by selecting it from the list.

Timer

A Timer control can execute code at regular intervals by causing a Timer event to occur. The Timer control, invisible to the user, is useful for background processing.

Picture Box

The primary use for the Picture Box control is to display a picture to the user. The actual picture that is displayed is determined by the picture property. The picture property contains the file name (and optional path) for the picture file that you wish to display.

DATA Grid Control

The DATAGrid control displays and operates on tabular data. It allows complete flexibility to sort, merge, and format tables containing strings and pictures. When bound to a Data control, MSFlexGrid displays read-only data.

Date and Time Picker Control

A Date And Time Picker (DTP) Control provides a simple and intuitive interface through which to exchange data and time information with a user. For example, with a DTP control you can ask the user to enter a data and then retrieve his or her selection with ease.

Option Button

An Option Button control displays an option that can be turned on or off.

Image Control

Use the Image control to display a graphic. An Image control can display a graphic from an icon, bitmap or metafile, as well as enhanced metafile, JPEG, or GIF files.

Check Box Control

A Check Box indicates whether a particular condition is on or off. We use check boxes in an application to give users true/false or yes/no options. Because check boxes work independently of each other, a user can select any number of check boxes at the same time.

References

1. *Kamal Acharya. School management system project report. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254873.34023165/v1*
2. *2. Kamal Acharya. A CASE STUDY OF CINEMA MANAGEMENT SYSTEM PROJECT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254873.30191075/v1*
3. *3. Kamal Acharya. A CASE STUDY ON ONLINE TICKET BOOKING SYSTEM PROJECT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254872.26972790/v1*
4. *4. Kamal Acharya. Web chatting application project report management system. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254871.18588592/v1*
5. *5. Kamal Acharya. RETAIL STORE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254871.14590154/v1*
6. *Kamal Acharya. SUPERMARKET MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252491.19145062/v1*
7. *Kamal Acharya. SOCIAL MEDIA MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252491.11210579/v1*
8. *8. Kamal Acharya. Online music portal management system project report. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252488.89734698/v1*
9. *Kamal Acharya. COLLEGE BUS MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172245277.70798942/v1*
10. *10. Kamal Acharya. AUTOMOBILE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172245276.67982593/v1*
11. *Kamal Acharya. Ludo management system project report. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172243999.98091616/v1*
12. *Kamal Acharya. Literature online quiz system project report. Authorea. July 31, 2024 DOI: https://doi.org/10.22541/au.172243825.53562953/v1*

13. *Kamal Acharya. CHAT APPLICATION THROUGH CLIENT SERVER MANAGEMENT SYSTEM PROJECT. Authorea. July 29, 2024. DOI: https://doi.org/10.22541/au.172228527.74316529/v1*
14. *15. Acharya, Kamal, Online Job Portal Management System (May 5, 2024). Available at SSRN: https://ssrn.com/abstract=4817534 or http://dx.doi.org/10.2139/ssrn.4 817534*
15. *16. Acharya, Kamal, Employee leave management system. (May 7, 2024). Available at SSRN: https://ssrn.com/abstract=4819626 or http://dx.doi.org/10.2139/ssrn.4 819626*
16. *17. Acharya, Kamal, Online electricity billing project report. (May 7, 2024). Available at SSRN: https://ssrn.com/abstract=4819630 or http://dx.doi.org/10.2139/ssrn.4 819630*
17. *18. Acharya, Kamal, POLICY MANAGEMENT SYSTEM PROJECT REPORT. (December 10, 2023). Available at SSRN: https://ssrn.com/abstract=4831694 or http://dx.doi.org/10.2139/ssrn.4 831694*